**LETT Software Upgrade: LETT → LETTX**

**WIJZIGINGEN**

**Cross Platform**

De vraag voor het herschrijven van de LETT applicatie kwam voort uit het probleem dat de studenten met een Mac geen gebruik konden maken van de software. Hierdoor moesten de beschikbare Windows computers worden gedeeld. Om dit probleem op te lossen is er gezocht naar een wijze om de applicatie op zowel Mac OS als Windows te laten draaien.

De gekozen oplossing was om over te stappen van het originele programma geschreven in Visual Basic naar een programma in Java. De programmeertaal is platform onafhankelijk en heeft de mogelijkheid om makkelijk externe bibliotheken toe te voegen.

**Simplificatie**

**Problemen:**

* Het originele programma vereiste een korte installatie. Dit betekend dat de applicatie vaak onnodig bewaard bleef na gebruik.
* Na de installatie waren er vaak problemen met het starten van de applicatie. Foutmeldingen waren het resultaat van het starten van de verkeerde link. De applicatie gaf aan dat het programma al geïnstalleerd was op een andere locatie zonder te specificeren wat deze locatie is.
* Daarnaast is het afhandelen van error ten gevolge van het verkeerd invoeren van de seriële poort of het ontbreken van een beschikbare seriële poort slordig en onduidelijk.
* Ten slotte is de interface van het programma enigszins verwarrend bij het eerste gebruik en niet heel aantrekkelijk.

**Oplossingen:**

* Java programma als JAR exporteren. Geen installatie vereist en gelijk te starten op beide platforms.
* Duidelijke communicatie over wat de situatie is betreffende de beschikbare seriële poorten. Opstellen van een overzicht met beschikbare poorten en duidelijke feedback vanuit applicatie.
* Versimpelen van de interface, waarbij er zoveel mogelijk wordt geprobeerd om de gebruiker bij eerste gebruik en zonder handleiding het programma te laten snappen.
* *Het toevoegen van fysieke knoppen op de LETT trekbanken helpt om de gebruiker makkelijker de proef goed af te laten stellen.*

**GEBRUIK**

**Installatie**

Het programma werkt zonder installatie op beide platforms. Echter wanneer er geen standaard Java JDK is geïnstalleerd of de huidige Java versie niet werkt met het programma, zal er een pop up verschijnen die de gebruiker linkt naar de Java download pagina. Hier moet de gebruiker zelfstandig de laatste JDK selecteren en instaleren *(Maken van instructies bij deze situatie).*

**Test Uitvoeren**

De gebruiker loopt van boven naar beneden chronologisch de stappen af om tot het eindresultaat te komen.

* Eerst zal een keuze moeten worden gemaakt welke poort te gebruiken.
* Vervolgens zal de locatie gewenste voor het te maken bestandje kunnen worden gekozen.
* Daarna zal er een naam voor het nieuwe bestand moeten worden gekozen.
* Dan zijn er drie stappen waarin de configuratie voor de trekproef kan worden gekozen. Eerst is er de keuze voor trek of compressie proef. Ten tweede de keuze voor de loadcell die wordt gekozen. Als laatste de snelheid waarmee de trekproef moet verlopen.
* Vervolgens bied de applicatie ook naast de fysieke knoppen op de LETT mogelijkheden om de hoogte van de grijper te verstellen zodat het specimen kan worden vastgezet.
* Ten slotte zal de gebruiker de test kunnen starten.

**Resultaat**

De resultaten van de test zullen in een .txt bestand worden geschreven op de locatie die de gebruiker heeft geselecteerd. In het tekstbestand verschijnt een header waarin alle informatie over de verrichte proef staat geschreven. Daarna een log waarin alle data stap voor stap staat geschreven. Deze data zal door de gebruiker kunnen worden verwerkt tot een Excel bestand waar de nodige data analyse over kan worden toegepast.

**PROGRAMMA**

**JSSC Library**

Om seriële communicatie op zowel Mac OS als Windows goed te laten verlopen is er gekozen om gebruik te maken van de JSSC library.

<https://github.com/scream3r/java-simple-serial-connector>

<https://code.google.com/archive/p/java-simple-serial-connector/>

De bibliotheek kan makkelijk met zowel de COM structuur van Windows als de TTY structuur van Mac OS voor de verwerking van seriële poorten omgaan. De Bibliotheek verwerkt zo alle communicatie tussen platform en Arduino zonder dat er platform specifieke code moet worden geschreven.

De bibliotheek wordt bijgehouden door de Gradle die via Maven Central de laatste versie van de bibliotheek opvraagt. Zo was het mogelijk de laatste versie van de bibliotheek te gebruiken. Mocht de bibliotheek ooit nog worden geüpdatet, dan zal de applicatie makkelijk naar de nieuwere versie om te zetten zijn.

**Structuur**

In de source van het programma is het programma opgebouwd uit de application, dao en gui.

**Application**

De LettxApplication class start de applicatie.

**DAO (Data Acces Object)**

In de SerialPortCommDao class wordt alle communicatie van en naar de Arduino geregeld.

* writeCommand() kan worden gebruikt voor het versturen van data.
* serialEvent() wordt getriggerd wanneer er data van de Arduino binnenkomt, waarna de data meteen wordt verwerkt binnen de event.
* initSerialPort() wordt gebruikt om de communicatie met de Arduino op te zetten.
* createTestLog() hiermee wordt het bestandje waarin alle data wordt geschreven aangemaakt.

**GUI**

Gui bestaat uit een form en een .java class. De form bevat de Java Swing code voor het creëren van de lay-out van het programma. De class zorgt voor het uitvoeren van de gui.

* GUI() voert de GUI uit wanneer het programma wordt gestart.
* initGui() initieert alle elementen van de GUI() zoals bijvoorbeeld de knoppen
* start() wordt uitgevoerd wanneer er op de start knop wordt gedrukt en loopt alle stappen door die nodig zijn voor het uitvoeren en verwerken van de proef.
* createUIComponents() initieert de custom created elementen van de UI.